



SC  
#10  
8/28/04

**PATENT**

**IBM Docket No. CA9-1998-0006**

Express Mail Label No. <u>E150316452305</u>	Date of Deposit: <u>08/05/2004</u>
I hereby certify that this paper and fee are being deposited with the United States Postal Service Express Mail envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria VA 22313-1450	
<u>PATSY SPEARS</u> Name of Person Mailing Paper and Fee	<u>Patsy Spears 08/05/2004</u> Signature of Person Mailing Paper and Fee

**In the United States Patent and Trademark Office**

---

**Date:** 5 August 2004

**In re Application** Andrew Blau et al.  
**of:**

**Filed:** 10/12/2000

**For:** System and Method for Managing Messages and Annotations  
Presented In A User Interface

**Serial Number:** 09/687,092

**Art Unit:** 2174

**Examiner:** Chuong, Truc

**SUPPLEMENTAL APPEAL BRIEF**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**RECEIVED**

AUG 10 2004

Sir:

Technology Center 2100

This is an appeal of a final rejection dated September 24, 2003 of Claims 1-30 of application serial number 09/687,092, filed 10/12/2000. This supplemental appeal brief is submitted pursuant to a communication mailed, 5/05/04, reopening prosecution and setting forth a new ground of rejection. Applicants respectfully request reinstatement of the appeal and submit this supplemental appeal brief in response thereto.

08/10/2004 AWONDAF1 00000007 090447 09687092

01 FC:1402 330.00 DA

**REAL PARTY IN INTEREST**

The application is assigned to International Business Machines, A Corporation of New York, Armonk, N.Y. 10504.

**RELATED APPEALS AND INTERFERENCES**

There are no related appeals and interferences of the claimed invention.

**STATUS OF CLAIMS**

The claims on appeal are Claims 1-30. No claims have been canceled. Each of Claims 1-30 has been finally rejected. The Examiner objected to Claims 6 and 23 as dependent upon a rejected base claim, but would be allowable if rewritten in independent form.

**STATUS OF AMENDMENTS**

No amendment has been filed subsequent to the final rejection.

**SUMMARY OF THE INVENTION**

As described on pages 1-28 of the Application, and as shown in Figures 1-19, the present invention provides for managing annotated messages, such as error messages generated during software code development in a graphical user interface. During application code development, once code is created, it must be

**PATENT**

**IBM Docket No. CA9-1998-0006**

compiled to determine if the code contains errors. If there are errors, the compiler will generate error messages to help the code developer diagnose the coding errors. Figure 1 illustrates the normal work flow for an application code developer. The code is written 40 and compiled 41. The compiler generates error messages 42 which are used by the code developer to evaluate and fix the errors/bugs 43. This is followed by a recompile 41 which is a new pass through the error identification 42 and correction steps 43 until the code is error free. Applicants' invention adds a new step to the process by providing user annotations to the compiler error messages 44. Programmers working with the same compilers over extended periods of time, frequently come to recognize certain error messages, and know what the likely cause and solution to the problems which caused the error messages. This information remains in the developer's mind unavailable for assisting other developers who encounter the same error messages. In addition, programmers recognize a smaller subset of error messages, and have their interpretations that are specific to the quirks of a particular project. Applicants' invention provides a means to allow other code developers to take advantage of this knowledge by allowing user annotations to compiler error messages. A second, initially empty, error message file is created and maintained for each error messages generated by the compiler. (See Specification, Page 13, Line 12). When a code developer adds a note to the compiler error message, the user's note is stored in the second error message file. Turning to Figure 2, compiler error messages 53 are textual material stored in a message catalog 50. The compiler error messages are identified and accessed by way of message keys 52 for display to a code developer in a window on a user's workstation. Message catalog name 51 is combined with message keys 52 to generate unique annotation dialog keys 54 which are used to access the annotations 56 for each compiler error message stored in the separate file. Applicants' invention allows this separate

annotation file to be transmitted or copied to another code developer's workstation and when the compiler generates the error message, the appropriate annotation will be displayed thus allowing multiple code developers to share experiences.

**ISSUES PRESENTED FOR REVIEW**

1. Whether the Examiner's rejection of Claim 13 under 35 U.S.C. Section 101 as directed to non-statutory subject matter is proper.

2. Whether the Examiner's rejection of Claims 1-12 under 35 U.S.C. Section 112, second paragraph, as failing to set forth the subject matter Applicants' regard as their invention is proper.

3. Whether the Examiner's rejection of Claims 1-5, 7-22 and 24-30 under 35 U.S.C. Section 103(a) as being unpatentable over Mueller (U.S. Pat. 5,673,390) in view of Hughes (U.S. Pat. 6,275,223), is proper.

**GROUPING OF CLAIMS**

With respect to the Examiner's rejection of Claim 13 under 35 U.S.C. Section 101, the claim stands alone.

With respect to the Examiner's rejection of Claims 1-12 under 35 U.S.C. Section 112, second paragraph, the claims stand together.

With respect to the Examiner's rejection of Claims 1-30 under 35 U.S.C. Section 103(a), the claims are grouped as follows:

1. Claims 1-5, 7-12, 16-22 and 24-27 stand together;
2. Claims 13-15 and 29-30 stand together;
3. Claim 28 stands alone.

ARGUMENT

35 U.S.C. Section 101

Claim 13 stands rejected under 35 U.S.C. Section 101. According to the Examiner, "Claim 13 is a system claim; however, it is missing a control component or a readable medium to perform the instructions as claimed." Applicants submit that Claim 13 is directed to statutory subject matter. Claim 13 recites "a first event driven control component for selecting from said first file and displaying a compiler error message from said first file in said user display." The specification recites at Page 16, Lines 12-17, that Applicants' invention operates with an event driven graphical user interface (GUI). The specification further recites that GUIs are event driven and are employed in the operation of the claimed invention. The specification further discloses a GUI generated user display as shown in Figures 9 and 10. There is no requirement that a claim contains a control component or a readable medium as stated by the Examiner. Therefore, the Examiner's rejection of Claim 13 under 35 U.S.C. Sec. 101 is erroneous, and should be reversed.

ARGUMENT

35 U.S.C. Section 112, second paragraph

Claims 1-12 stand rejected under 35 U.S.C. Sec. 112, second paragraph. According to the Examiner,

"Evidence claimed in claims 1, 11, and 12 fails to correspond in scope with that which applicant(s) regard as the invention .... for example, in Claim 1 lines 4-5, the Applicant claimed "associating said annotation with said compiler error message using a unique key an storing said annotation in said separate empty error file (storing the annotation in the same file which stores the compiler error message)"

The Examiner concluded that Applicants' claims recite storage of the annotation and the compiler error messages in one file. Applicants submit that Claims 1, 11, and 12 are completely

**PATENT**

**IBM Docket No. CA9-1998-0006**

clear and particularly point out and distinctly claims the subject matter that Applicants regard as the invention. Claims 1, 11, and 12 recite "displaying a compiler error message having a separate empty error file to a user". Webster's Ninth New Collegiate Dictionary, Copyright 1983, defines the word "separate" as 1) detached; 2) not shared with another; 3) existing by itself. The claim is clearly reciting that the compiler error message has a "detached", "not shared", "existing by itself" empty error file. Applicants' claims further recite that the annotation is associated with the compiler error message using a unique key and that the annotation is stored in the separate empty error file. Applicants are at a loss to understand how anyone, with or without having read Applicants' disclosure and with or without any skill in any art, could fail to understand the claimed recitation of two separate files and NOT one file. Therefore, the Examiner's rejection of Claims 1-12 under 35 U.S.C. Sec. 112, second paragraph is erroneous, and should be reversed.

**ARGUMENT**

**35 U.S.C. Section 103(a)**

Claims 1-5, 7-22 and 24-30 stand rejected under 35 U.S.C. Section 103(a) as being unpatentable over Mueller in view of Hughes. That rejection is not well founded and should be reversed.

The Examiner's rejection of the claims is erroneous for several reasons. The combination suggested by the Examiner is not fairly suggested by the references. Moreover, the references cannot be combined to achieve Applicants' invention.

The Hughes reference discloses a code inspection tool for allowing developers to compile annotation data and forward the annotation data from a graphical user interface to a centralized

data store (See Hughes, Col. 3, Lines 57-60). Hughes discloses during code inspection sessions, that all developers view a display, which sets out original source code, side by side with new source code on a line by line basis (See Hughes, Col. 3, Lines 62-67), matching the line numbers of the original source with those of the new source code. Annotations to the source code lines are indicated by further icons next to the appropriate lines (See Hughes, Col. 4, Lines 3-5). Hughes fails to show annotations for errors generated from compiled source code, storage of those annotations in a separate file, or the use of a unique key to store the annotations in the separate file as shown in Applicants' invention.

Mueller discloses a technique for interactively displaying error messages, such as parser or compiler messages, associated with a user's source code. Mueller discloses that compilers typically provide programmers with a compiler listing which lists the source code along with the errors (See Mueller, Col. 1, Lines 29-31). Mueller teaches that these errors may be listed at the end of the source listing, interspersed through the source listing, or in some instances the errors are displayed in the same window in which the source file is being edited (See Mueller, Col. 1, Lines 31-36). Mueller specifically address special problems where compiler errors code emanate from multiple sources, such as a local parser or compiler, as well as a remote compiler which runs on a second computer, such as a mainframe host computer connected through a communication link (See Mueller, Col. 1, Lines 52-59).

The subject matter of Claim 1, taken as a whole, is neither taught nor suggested by any combination of the references relied upon by the Examiner. Specifically, neither of the reference teaches or suggests displaying a compiler error message having a separate empty error file into which an annotation is associated and stored. Hughes is directed to source code inspection which is far different from compiler error examination. The reasons

why such a feature cannot be implemented in Hughes is the fact that it would require an empty error file for each line of source code, which would require enormous storage requirements.

Applicants' invention in contrast is working only with the errors generated from compiler error diagnosis which results in limiting annotations and storage to a limited set of errors. Mueller also fails to disclose this feature of Applicants' invention. Mueller is directed to the gathering of error messages from multiple locations and displaying those errors in response to a user's request. There is no teaching in either reference that would cause one skilled in the art to make Applicants' claimed invention. Similarly, neither of the references discloses associating an annotation with a compiler error message using a unique key as disclosed in Applicants' invention. Applicants' invention disclose that compiler error messages are identified and accessed by way of message keys 52 (Figure 2) for display to a code developer in a window on a user's workstation. Message catalog name 51 (Figure 2) is combined with message keys 52 to generate unique annotation dialog keys 54 which are used to access the annotations 56 for each compiler error message stored in a separate file. There is no teaching in either of the references that would suggest the generation of a unique annotation dialog key as in Applicants' invention. Accordingly, the Examiner's rejection of Claim 1 is erroneous and should be reversed. Claims 2-5, 7-12, 16-22 and 24-27 stand or fall with Claim 1.

With respect to Claim 13, neither of the references teaches an event driven control component for displaying using a unique key an associated annotation in said user display. In Hughes, the source code line numbers are used to associate an annotation with a particular line of source code. Mueller requires that the error message itself, and not an annotation, be stored such that the type and location maybe determined. Neither of the references disclose the use of a unique key for displaying an



**PATENT**

**IBM Docket No. CA9-1998-0006**

associated annotation to a user. Accordingly, the Examiner's rejection of Claim 13 is erroneous and should be reversed. Claims 14-15 and 29-30 stand or fall with Claim 13.

With respect to Claim 28, there is no teaching or suggestion in any of the references of the concept of selectively displaying an annotation with a compiler error message and causing a computer to effect presenting an edit panel in said graphic user interface for user entry of new or modified annotations. Accordingly, the Examiner's rejection of Claim 28 is erroneous and should be reversed. Claim 28 stands alone.

**CONCLUSION**

It is therefore respectfully urged that the Examiner's rejection of Claim 13 under 35 U.S.C. Section 101 is erroneous and that the Examiner's rejection of Claims 1-12 under 35 U.S.C. Section 112, second paragraph, is likewise erroneous and that the Examiner's rejection of Claims 1-5, 7-22 and 24-30 under 35 U.S.C. Section 103(a) is still further erroneous. Accordingly, it is respectfully urged that the Examiner's rejections be reversed in all respects.

Please charge the \$ 330.00 fee, if required, for submission of a brief in support of appeal to Deposit Account No. *09-0447*. No additional fee is seen to be required. Please charge any excess or credit any overpayment to our Deposit Account No. *09-0447*.

Respectfully submitted,

*David A. Mims, Jr.*

David A. Mims, Jr.-Attorney

Registration No. 32,708

**PATENT**  
**IBM Docket No. CA9-1998-0006**

(512) 823-0950

APPENDIX

1. A method for managing compiler error messages, comprising the steps of:
  - displaying a compiler error message having a separate empty error file to a user;
  - accepting from said user an annotation to said compiler error message;
  - associating said annotation with said compiler error message using a unique key and
  - storing said annotation in said separate empty error file; andthereafter selectively displaying said annotation with said compiler error message.
2. The method of claim 1, said selectively displaying step further comprising the steps of:
  - displaying with said compiler error message indicia representing the existence of said annotation;
  - selectively receiving from said user a request to display said annotation; anddisplaying said annotation with said compiler error message.
3. The method of claim 1, said accepting step further comprising the steps of:
  - selectively presenting to said user an edit panel; and
  - receiving from said user said annotation input associated with said compiler error message to said edit panel.
4. The method of claim 1, further comprising the steps of:

selecting a compiler error message having a first key from a first file of compiler error messages for display to said user; and

associating in a second file said annotation to said selected compiler error message in said first file.

5. The method of claim 4, further comprising the steps of:

providing compiler error message identifying indicia for each compiler error message in said first file;

generating annotation identifying indicia as a function of said compiler error message identifying indicia.

6. The method of claim 4, further comprising the steps of:

upon presenting a compiler error message from said first file, determining the presence of a corresponding annotation in said second file using said second key;

responsive to the presence of said corresponding annotation,

displaying with said compiler error message indicia representing the existence of said annotation;

selectively receiving from said user a request to display said annotation; and

responsive to receiving the request from said user, displaying said annotation with said compiler error message using said second key.

7. The method of claim 1, further comprising the steps of:
  - during processing of application code entered by a user, identifying a compiler error in said code;
  - selecting and presenting to said user a compiler error message;
  - identifying and presenting to said user an annotation corresponding to said compiler error message; and
  - enabling and selectively receiving said annotation and a modified annotation from said user for association with said compiler error message.
8. The method of claim 7, further comprising the steps of:
  - preserving a history of compiler error messages presented to said user;
  - enabling user selection one of said compiler error message from said history of compiler error messages; and
  - selectively receiving from said user an annotation to the compiler error message selected from said history.
9. The method of claim 1, further comprising the step of presenting said annotation to other users receiving said compiler error message.
10. The method of claim 4, further comprising the step of enabling access by other users to said second file containing said annotations associated with said compiler error messages.

11. A method for managing compiler error messages in a graphic user interface, comprising the steps of:
  - selecting and displaying a compiler error message having a separate empty error file to a user;
  - accepting from said user an annotation to said compiler message;
  - associating said annotation with said compiler error message and storing said annotation in said separate empty error file using a unique key;
  - thereafter selectively displaying said annotation with said compiler error message; and
  - presenting an edit panel in said graphic user interface for user entry of new or modified annotations.
12. A system for managing compiler error messages at a user interface, comprising:
  - means for displaying a compiler error message having a separate empty error file to a user;
  - means for accepting from said user an annotation to said compiler error message;
  - means for associating said annotation with said compiler error message and storing said annotation in said separate empty error file using a unique key; and
  - means for selectively displaying said annotation with said compiler error message.
13. A system for presenting compiler error messages in a user display, comprising:

a first file for storing a plurality of compiler error messages, each said compiler error message identified by a message key;

a second file for storing a plurality of annotations, each said annotation associated with a corresponding one of said compiler error message;

a first event driven control component for selecting from said first file and displaying a compiler error message from said first file in said user display;

a second event driven control component for determining the presence in said second file of an annotation associated with said displayed compiler error message; and

a third event driven control component for displaying using a unique key said associated annotation in said user display.

14. The system of claim 13, further comprising a fourth control component responsive to entry in said user display of a annotation to a displayed compiler error message, for adding said annotation to said second file using a unique key associated with said displayed compiler error message.

15. (Amended) The system of claim 14, further comprising an editor for receiving via an annotation panel in said user display said annotation.

16. A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform the method steps for managing compiler error messages of claim 1.

17. A computer program product or computer program element for managing a compiler error message display according to claim 1.

18. An article of manufacture comprising:

a computer useable medium having computer readable program code means embodied therein for managing compiler error messages, the computer readable program means in said article of manufacture comprising:

computer readable program code means for causing a computer to effect displaying a compiler error message to a user having a separate empty error file;

computer readable program code means for causing a computer to effect accepting from said user an annotation to said compiler error message;

computer readable program code means for causing a computer to effect associating said annotation with said compiler error message and storing said annotation in said separate empty error file using a unique key; and

computer readable program code means for causing a computer to effect thereafter selectively displaying said annotation with said compiler error message.

19. The article of manufacture of claim 18, said computer readable program code means for causing a computer to effect selectively displaying further comprising:

computer readable program code means for causing a computer to effect displaying with said compiler error message indicia representing the existence of said annotation;



computer readable program code means for causing a computer to effect selectively receiving from said user a request to display said annotation using a unique key; and

computer readable program code means for causing a computer to effect displaying said annotation with said compiler error message.

20. The article of manufacture of claim 18, said computer readable program code

means for causing a computer to effect accepting further comprising:

computer readable program code means for causing a computer to effect selectively presenting to said user an edit panel; and

computer readable program code means for causing a computer to effect receiving from said user said annotation input to said edit panel associated with one of said compiler error messages.

21. The article of manufacture of claim 18, further comprising:

computer readable program code means for causing a computer to effect selecting a compiler error message from a first file of compiler error messages for display to said user; and

computer readable program code means for causing a computer to effect associating in a second file said annotation to a corresponding compiler error message in said first file.

22. The article of manufacture of claim 21, further comprising:

computer readable program code means for causing a computer to effect providing compiler error message identifying indicia for each compiler error message in said first file;

computer readable program code means for causing a computer to effect generating annotation identifying indicia as a function of said compiler error message identifying indicia.

23. The article of manufacture of claim 21, further comprising:

computer readable program code means for causing a computer to effect, upon presenting a compiler error message from said first file, determining the presence of a corresponding annotation in said second file;

computer readable program code means for causing a computer to effect, responsive to the presence of said corresponding annotation, displaying with said compiler error message indicia representing the existence of said annotation;

computer readable program code means for causing a computer to effect selectively receiving from said user a request to display said annotation; and

computer readable program code means for causing a computer to effect responsive to receiving the request from said user, displaying said annotation with said compiler error message.

24. The article of manufacture of claim 18, further comprising:

computer readable program code means for causing a computer to effect, during processing of application code by a compiler entered by a user, identifying a compiler error in said code;

computer readable program code means for causing a computer to effect, selecting and presenting to said user an error message corresponding to said error;

computer readable program code means for causing a computer to effect identifying and presenting to said user an annotation corresponding to said compiler error message; and

computer readable program code means for causing a computer to effect enabling and selectively receiving said annotation and a modified annotation from said user for association with said compiler error message.

25. The article of manufacture of claim 24, further comprising:

computer readable program code means for causing a computer to effect preserving a history of compiler error messages presented to said user;

computer readable program code means for causing a computer to effect enabling user selection of one of said compiler error message from said history of compiler error messages; and

computer readable program code means for causing a computer to effect selectively receiving from said user an annotation to the compiler error message selected from said history.

26. The article of manufacture of claim 18, further comprising computer readable program code means for causing a computer to effect presenting said annotation to other users receiving said compiler error message.
27. The article of manufacture of claim 21, further comprising computer readable program code means for causing a computer to effect enabling access by other users to said second file containing said annotation associated with said compiler error file.
28. An article of manufacture comprising:
- a computer useable medium having computer readable program code means embodied therein for managing compiler error messages in a graphic user interface, the computer readable program means in said article of manufacture comprising:
    - computer readable program code means for causing a computer to effect selecting and displaying a compiler error message having a separate empty error file to a user;
    - computer readable program code means for causing a computer to effect accepting from said user an annotation to said compiler error message;
    - computer readable program code means for causing a computer to effect associating said annotation with said compiler error message and storing said annotation in said separate empty error file using a unique key;
    - computer readable program code means for causing a computer to effect thereafter selectively displaying said annotation with said compiler error message; and

computer readable program code means for causing a computer to effect presenting an edit panel in said graphic user interface for user entry of new or modified annotations.

29. A computer program product for presenting compiler error messages in a user display, comprising:

a first file for storing a plurality of compiler error messages, each said compiler error message identified by a message key;

a second file for storing a plurality of annotations, each said annotation associated with a corresponding one of said compiler error message;

a first event driven control component for selecting from said first file a compiler error message for presentation in said user display;

a second event driven control component for determining the presence in said second file of an annotation associated with said compiler error message; and

a third event driven control component for displaying said associated annotation in said user display.

30. The computer program product of claim 29, further comprising a fourth control component responsive to entry in said user display of an annotation to a displayed compiler error message, for adding said annotation to said second file associated with said displayed compiler error message

**PATENT**

**IBM Docket No. CA9-1998-0006**

presenting an edit panel in said graphic user interface for user entry of new or modified annotations.